U.S. Department of State, Information Resource Management,

Office of Information Assurance, February 14, 2011

The following narrative highlights the constraints that were faced when building the State Department I-Post risk score manager. These statements which were developed by talking to Dr. Ron Rudman who wrote most of the code, serve as black box warnings to consider when evaluating the State Department risk score manager for wider use in the public or private sector.

The State Department is in the process of collecting the actual software code in I-Post to release to the Department of Defense. In a parallel effort my organization (IRM/IA) is reviewing how to provide additional support to external organizations that seek to adopt risk scoring as indicated in the later pages of this document. This effort is unfunded.

In the meantime, a run time version of software that simulates some of the features and data elements in I-Post is available on request. This disk called PRSM can be obtained by sending an e-mail to DOSCISO@state.gov. Please include a street address.

John Streufert

Chief Information Security Officer

US Department of State

Summary:

iPost is the custom GOTS application that continuously monitors and scores risk in the IT infrastructure at the Department of State (DoS). Subsequent to the Department's Risk Scoring program winning the National Security Agency's 2009 Frank B. Rowlett award for Organizational Achievement, there have been numerous requests for iPost so that other agencies could use it directly or as a guide to developing their own Risk Scoring program.

There are several factors other agencies should consider when trying to decide whether to adopt iPost (or to build/buy an alternative):

- iPost includes data (and database objects) to support network operations management, not just Risk Scoring alone.
- iPost was developed with a limited budget to meet State requirements. As such it has features/concepts that may not translate well for other agencies. (In other words, it was not an original design requirement that iPost would be "generic" in ways that would facilitate export of this tool to other organizations.)
- iPost was never intended to be used outside of DoS and does not have either the support for customization or the type of documentation that would be ideal for a product intended to be "off-the-shelf."
- As a result of the last bullets, other agencies should consider whether any investment to adapt iPost to their needs might be better spent starting over with PRSM (see below) or a commercial tool.

An alternative to iPost for adoption by other agencies is PRSM.

- This system eliminates features not related to risk scoring.
- It also lacks operational code to load data, and has a much less nuanced user interface.
- The data structures are still too State-centric.
- It has relatively complete documentation.

iPost is going thru an upgrade from version 3.3 to version 3.4 due to release at the end of May 2010. The new release will include the following major changes:

- Scoring changes (calculation of site score as the total of the averages of the component scores, ability to override score for individual patches, etc.)
- LAN topology diagrams
- FSI training expiration dates (e.g. Cyber Security awareness training)
- Search Box
- User authorization from the database
- Redesigned home page that includes navigation

Considering the new changes provide major improvement to iPost, it is recommended that the effort concentrate on exporting iPost 3.4 artifacts.

Objectives:

In order to provide the appropriate support to external customers, IA is looking to build the expertise in-house in order to address any support and follow-on questions.

IA plans to install, configure and implement iPost on a standalone system, with minimal support from the iPost team. This will help gauge and document the level of effort for any other Federal Department or Agency to implement iPost as well.

iPost Components:

Databases:

- SCRUB database- Raw data from all data sources first populated in this database. This database is used to reconcile the data prior to use by iPost.
- STAGE database – Once data is reconciled, the pertinent data is populated in the STAGE database.
- PROD database – Used for all reports and screens. This majority of data in this database is populated via SQL Server replication every 20 minutes from the stage database.
- iPost Data Warehouse – This database houses historical iPost data.

- Partitions – Used only for the network features in iPost such as OpenView, Tavve data. There's a significant amount of data in this database as its keeping the history of all information (e.g. Tavve data for the past quarter).

A rough estimate of the storage space consumed by all the above mentioned databases is approximately 1/2 Terabyte. If replication is not used, this size would be further reduced.

Front end:

The front-end GUI for iPost is written in C# and uses Microsoft .Net as the framework. The code is stored in a repository with folder structure that groups parts of the code that are inter-related. There is no documentation on the code directory structure or the relationship between the files.

Reports:

The reports are created with SQL Server Reporting Services. There are approximately 75-100 reports and they are in XML format. There's a command line tool in Reporting Services that imports the .rdl file formats of the exported reports into a separate SQL Server database.

Note: If the web server is installed on OpenNet it will have to be on a separate server than the database server (per DS guidance). The reporting service can be installed on the same server as the web server – the database that the rdl files are imported into would reside on the same server as the other databases.

Exclusions:

The following data is not kept in the iPost database. It is retrieved directly from the data sources:

- RCSO Report
- SMS Advertisements
- Remedy Tickets
- UTT Tickets
- Tenable drill through to how-to-fix-it details

- NetOMNI (Network Usage)

Level of Effort:

Databases – There are several options for obtaining the databases from ENM:

1. Restore backups of all the iPost databases – The iPost team would create backups of all of the iPost databases, which can then be restored via SQL Server Management Studio.
2. Extract the database schemas only – The amount of work for this option would be less but would yield only the schemas and no data/content. If this option is selected, a subset of the data would need to be exported for IA team's understanding of the content.

Exporting the databases to an external format is not a viable option because there is much additional work and time involved and is really unnecessary. Approximate time for the iPost team to complete any of the options listed would be 3-5 days.

Front-end – The code for the GUI could be provided by exporting the code from the repository into Visual Studio projects. The approximate time to complete would be approximately 1-2 days.

IA would need to create AD accounts, update the registry keys that the front-end uses, and grant these accounts access to SQL. The iPost team would provide a utility to encrypt the connection strings that are stored on the web server. The easiest approach would be to the GUI and database on the same server using impersonation.

Reporting – The reports would be exported as Visual Studio projects.

Scheduled tasks – There are approximately 50 scheduled tasks that would be exported. The exported data would have to be modified to reflect proper file system locations, hostname changes, and to point at the IA installation of SQL Server Integration Services.

Roles and responsibilities:

ENM iPost team:

- Complete the necessary work to transfer the iPost software to IA.
- Provide “forum” type support (i.e. respond to questions as time permits).

IA team:

- Provide method for transfer of the iPost software (e.g. Share folder, etc.)
- Install the iPost software on a standalone system for learning/evaluation purposes.
- Build competency in iPost software to provide the initial support for external customers.

Assumptions:

- IA team will be assessing the complexity of the installation and support of the iPost software. As such the iPost team will not be directly involved in the process.
- The effort will be for transferring version 3.4 of iPost software due for release May 31, 2010.

Constraints:

- iPost development timeline is driven by internal customer requirements and constrained by resources.
- Currently iPost team resources are 100% dedicated to meeting the next release version 3.4.
- There will be a 6-12 month learning curve with any new resources added to the iPost team.
- The IA installation of iPost software would have to go through C&A if it is implemented on OpenNet.
- The data sources will not be connected to the IA iPost installation for real-time feed unless IA coordinates directly with the data owner (e.g. SMS, Tenable, etc.).

- Other initiatives could impact availability of iPost team (e.g. DS transition to McAfee).